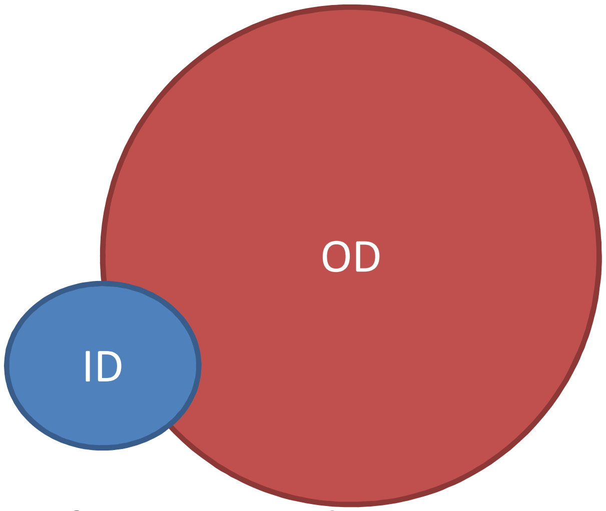
Domain Adaptation Using Joint Models

Nadir Durrani

Team




Domain Adaptation



- Preserve lexical choice, writing style, reordering of In-domain genre
- Making best use of the additional general/out-domain data to improve the performance of the system on in-domain task
- Model Weighting
- Data Selection

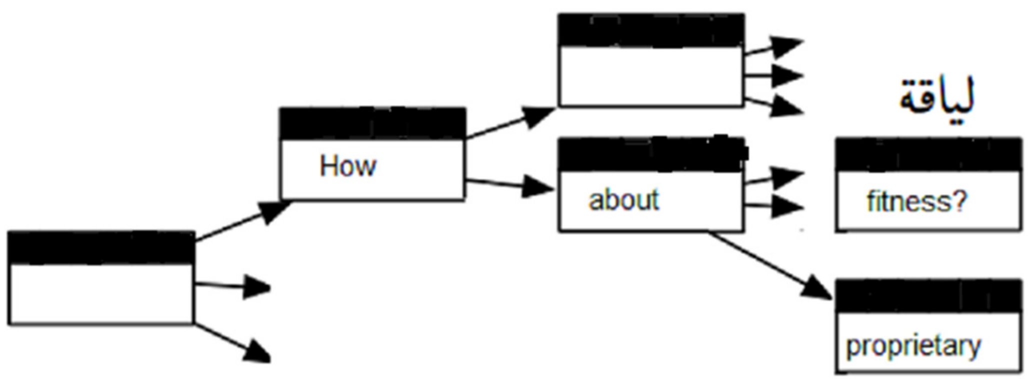
Model Weighting

- Skew the probability distribution towards ID
  - Concatenate ID data multiple times
  - Weighted interpolation, instance weighting
  - Domain indicator features

Model Weighting

- Skew the probability distribution towards ID
  - Concatenate ID data multiple times
  - Weighted interpolation, instance weighting
  - Domain indicator features

ماذا عن لياقة البدن؟



Data Selection

- Select pseudo-domain data from the out-domain data
  - Train system on In-domain + Concatenation
- Train in and out-domain models
  - Select data based on cross-entropy difference
- Pros
  - Important for speed and memory (SAC Demo)
- Cons
  - Cumbersome to find optimal threshold
  - System cannot fallback to general domain

NNJM Model

- Neural Network Joint Model – NNJM (Devlin et. al 2014: ACL Best Paper)

- Augments language model with source context window

$$P(T|S) \approx \Pi_{i=1}^{|T|} P(t_i | t_{i-1}, \cdots, t_{i-n+1}, \mathcal{S}_i)$$

عن مشكلة الحمل الزائد للاختيار

About the problem of choice overload

- p(problem | the, About, <s> عن, مشكلة, الحمل)
  - Typically 14-gram model (9 source words + 5 target) is used

Why NNJM?

- Handles source and target contextual dependencies across phrasal boundaries
- The n-gram units capture reordering patterns
- Capture semantic dependencies and generalized information
- Gave an improvement of +3.0 BLEU points on top of top ranked Arabic-English NIST system
- Already implemented in Moses and part of SOTA pipeline

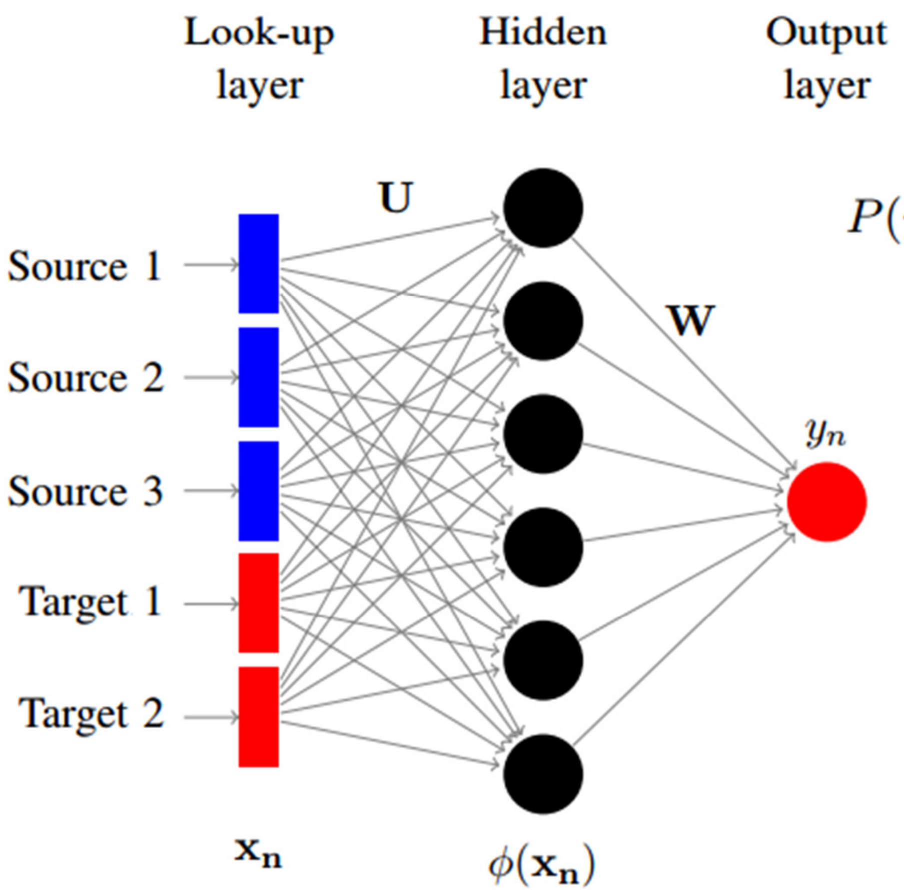
Motivation

- Hypothesis: An NNJM trained on plain concatenation of in- and out-domain data is suboptimal

- Can we learn model in a way that it prefers in-domain?
  - NDAM: Instance weighting by regularizing the loss function
  - NFM: Fusion of in- and out-domain models

- Can we do data selection using NNJM?

NNJM Model



$$P(y_n = k|\mathbf{x_n}, \theta) = \frac{exp\left(\mathbf{w}_k^T \phi(\mathbf{x_n})\right)}{\sum_{m=1}^{|V_o|} exp\left(\mathbf{w}_m^T \phi(\mathbf{x_n})\right)}$$

$$J(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{|V_o|} y_{nk}\, log\, P(y_n = k|\mathbf{x_n}, \theta)$$

NDAM Models – EMNLP ‘16

- NNJM Model trained on plain concatenation might be suboptimal

- We train model on weighted concatenation
  - NDAM-v1: Drift towards out-domain model is controlled using regularizer based on in-domain model
  - NDAM-v2: Regularizer is based on cross entropy difference of in- and out-domain model

NDAM Models – EMNLP '16

- NDAM-v1:

$$J(\theta_a) = \sum_{n=1}^{N} \sum_{k=1}^{|V_o|} \Big[\lambda \, y_{nk} \, logP(y_n = k|\mathbf{x_n}, \theta_a) + (1 - \lambda)$$

$$y_{nk} \, P(y_n = k|\mathbf{x_n}, \theta_i) \, logP(y_n = k|\mathbf{x_n}, \theta_a)\Big]$$

- NDAM-v2:

$$J(\theta_a) = \sum_{n=1}^{N} \sum_{k=1}^{|V_o|} \Big[\lambda \, y_{nk} \, logP(y_n = k|\mathbf{x_n}, \theta_a) + (1 - \lambda)$$

$$y_{nk}\Big[P(y_n = k|\mathbf{x_n}, \theta_i) - P(y_n = k|\mathbf{x_n}, \theta_a\Big] \, logP(y_n = k|\mathbf{x_n}, \theta_a)\Big]$$

Technical Issues

- Handling OOVs
  - Probability of sequences containing OOV is high according to the ID model
  - Mark out-domain OOVs by OOV_o
- Vanishing and Exploding Gradients
  - Gradient clipping [+5,-5]
- NCE to avoid repetitive softmax computation
  - For each training instance, sample 100 samples
  - Unigram versus Uniform
  - NCE loss is defined to discriminate from true instance from noisy ones

Fusion Models (COLING' 2016)

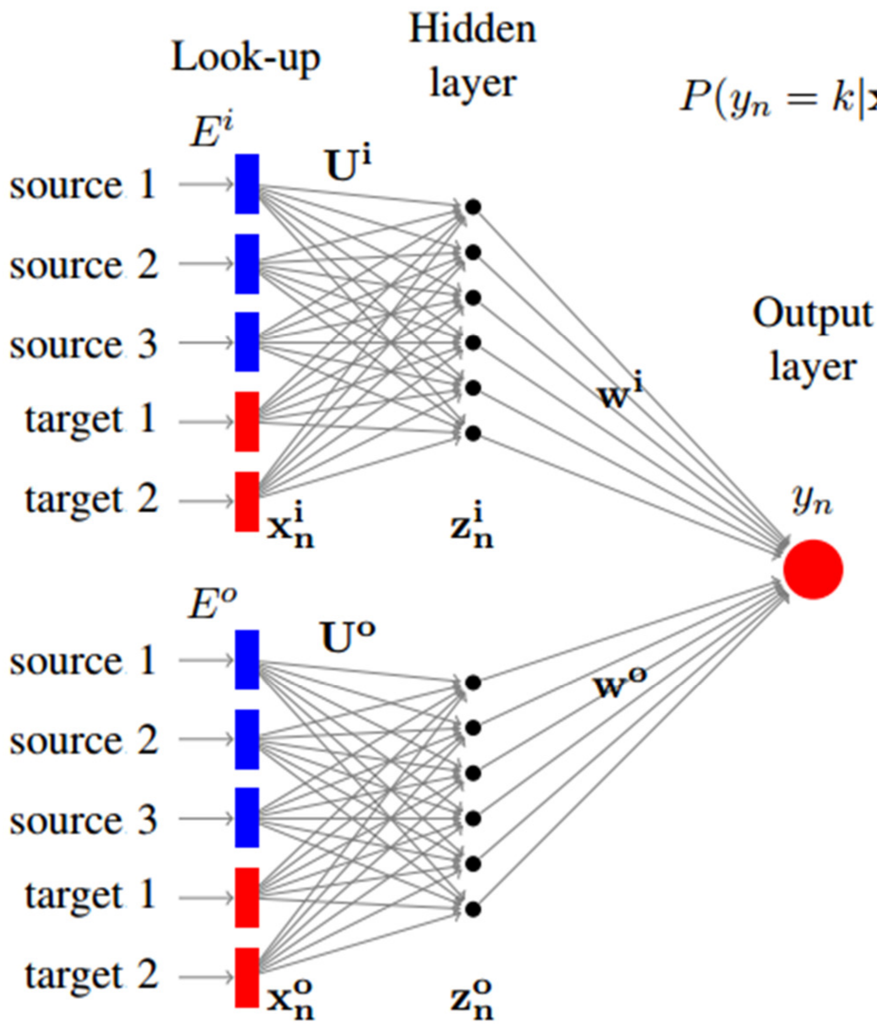
- Motivation: Interpolation of language model, phrase-tables by minimizing perplexity

- Interpolating models to minimize perplexities on tune

$$P(\mathbf{x}_n|\theta,\lambda) = \sum_{d=1}^{D} P(\mathbf{x}_n|z_n = d, \theta_d)\ \lambda_d$$

- NFM: Train in- and out-domain models separately, train a composite model
  - Use in-domain data to back-propagate errors
  - Adjust the weights of embedding and outer layers

Fusion Model



$$P(y_n = k|\mathbf{x_n^i}, \mathbf{x_n^o}, \theta^i, \theta^o) = \frac{\exp\ ([\mathbf{w}_k^i, \mathbf{w}_k^o]^T[\mathbf{z}_n^i, \mathbf{z}_n^o])}{\sum_{m=1}^{|V_o|} \exp\ ([\mathbf{w}_m^i, \mathbf{w}_m^o]^T[\mathbf{z}_n^i, \mathbf{z}_n^o])}$$

$$\nabla_{\mathbf{w}_j^d} J(\theta) = \sum_{n=1}^{N} \Big[(y_{nj} - \sigma_{nj})\mathbf{z}_n^d \Big]$$

Data and Settings

- Language Pairs: Arabic, German
- Data: In-domain: TED, Out-domain: UN (AR), EP, CC, News
- NNJM Settings: Vocab [20K, 40K], word vector size D 150, hidden layer 750, SGD with NCE using 100 noise samples
- Baseline: Moses with SOTA features and settings
- Tune IWSLT dev and test-10-, Test IWSLT[11-13]

Results (Adapting NNJM Models)

	German		Arabic	
System	Avg	Δ	Avg	Δ
Baseline (NNJM)	24.9		28.7	
NDAM	25.3	+0.4	28.9	+0.2
Linear	25.3	+0.4	29.1	+0.4
Log-Linear	25.3	+0.4	29.0	+0.3
Fine Tuning	25.6	+0.7	28.9	+0.2
NFM-I	25.8	+0.9	29.4	+0.7
NFM-II	25.6	+0.7	29.2	+0.5

Results (Phrase-table Adaptation)

	German		Arabic	
System	Avg	Δ	Avg	Δ
Baseline (NNJM)	24.9		28.7	
PT Interpolation	25.2	+ 0.3	28.9	+0.2
Instance Wt.	25.3	+ 0.4	29.2	+0.5
Fill Up	25.1	+ 0.2	28.9	+0.2
NFM-I	25.8	+0.9	29.4	+0.7
NFM-I + Instance Wt.	25.7	+0.8	29.7	+1.0

Results (Data Selection)

	German	Arabic
System	Avg	Avg
Baseline$_{cat}$	24.9	28.7
Baseline$_{ID}$	24.3	29.1
MML	24.7	29.7
+NFM-I	25.2	30.0

Summary

- Novel Domain Adaptation Models based on the NNJM model
  - NDAM models: regularizing loss function based on cross-entropy difference
  - Fusion Models: combining in- and out-domain model through back-propogation
- Applied known techniques
  - Linear interpolation
  - Log-linear interpolation
  - Data Selection using NNJM

Summary

- Fusion models performed best among the methods also beating phrase-table adaption

- We found methods to be complementary
  - Gains on top of phrase-table adaptation and data selection

References

Shafiq Joty, Nadir Durrani, Hassan Sajjad, and Ahmed Abdelali. 2017. Domain Adaptation using Neural Network Joint Model, Computer Speech & Language (2017)

Nadir Durrani, Hassan Sajjad, Shafiq Joty, Ahmed Abdelali, and Stephan Vogel. 2016. A Deep Fusion Model for Domain Adaptation in Phrase-based MT. In Proceedings of the Twenth Sixth Annual Conference on Computational Linguistics (COLING), Osala, Japan, December.

Nadir Durrani, Hassan Sajjad, Shafiq Joty, Ahmed Abdelali, and Stephan Vogel. 2015. Using joint models for domain adaptation in statistical machine translation. In Proceedings of the Fifteenth Machine Translation Summit (MT Summit XV), Florida, USA, October. AMTA

Shafiq Joty, Hassan Sajjad, Nadir Durrani, Kamla Al-Mannai, Ahmed Abdelali, and Stephan Vogel. 2015. How to Avoid Unwanted Pregnancies: Domain Adaptation using Neural Network Models. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, September.